WHAT IS CLAIMED IS:

1. A processor comprising:

an execution unit configured to execute one or more threads; and

a detection unit coupled to detect whether a given thread includes an identifier;

wherein said execution unit is further configured to selectively continue execution of said given thread depending upon whether said detection unit detects said identifier.

2. The processor of claim 1, wherein in response to said detection unit detecting said identifier, said execution unit is configured to suspend execution of said given thread and to execute a different thread.

3. The processor of claim 2, wherein in response to said detection unit detecting that said given thread does not include said identifier, said executing unit is configured to continue execution of said given thread.

4. The processor of claim 1, wherein said execution unit is configured to suspend execution of said given thread and to execute a different thread in response to receiving a global execution parameter.

5. The processor of claim 4, wherein in response to said detection unit detecting said identifier, said execution unit is configured to override said global execution parameter and to continue execution of said given thread.

6. The processor of claim 1 further comprising a priority designation unit coupled to said detection unit and configured to assign a priority level to said given thread depending upon an execution environment in response to said detection unit detecting said identifier.

7. The processor of claim 6, wherein in response to said given thread having a priority level lower than said different thread, said execution unit is configured to suspend execution of said given thread and to execute said different thread with a higher priority level

8. The processor of claim 7, wherein in response to said given thread having a priority level the same as or higher than said different thread, said execution unit is configured to continue execution of said given thread.

9. The processor of claim 1, wherein said identifier is a unique instruction.

10. The processor of claim 1, wherein said identifier is a flag including one or more unused bits of any instruction of said given thread.

11. The processor of claim 1, wherein said identifier is information associated with any instruction of said given thread.

12. A method comprising:

executing one or more threads;

detecting whether a given thread includes an identifier; and selectively continuing execution of said given thread depending upon whether said identifier is detected.

13. The method of claim 12, wherein in response to detecting said identifier, suspending execution of said given thread and executing a different thread.

14. The method of claim 13, wherein in response to detecting that said given thread does not include said identifier, continuing execution of said given thread.

15. The method of claim 12, further comprising suspending execution of said given thread and executing a different thread in response to receiving a global execution parameter.

16. The method of claim 15, wherein in response to detecting said identifier, overriding said global execution parameter and continuing execution of said given thread.

17. A method of generating low-level instructions executable by a processor, said method comprising:

providing a computer program including high-level programming instructions;

detecting whether an indicator is included within said computer program; and

in response to detecting said indicator, generating a low-level instruction having an identifier corresponding to said indicator, wherein said low-level identifier is configured to cause said processor to selectively continue execution of a given thread.

18. The method of claim 17, wherein said indicator is a compiler directive.

19. The method of claim 17, wherein said indicator is an assembly language subroutine.

20. The method of claim 17, wherein said indicator is a unique high-level instruction.

21. A machine-readable medium comprising program instructions, wherein said program instructions are executable by a processor to:

execute one or more threads;

detect whether a given thread includes an identifier; and

selectively continue execution of said given thread depending upon whether said identifier is detected.

22. A processor comprising:

means for executing one or more threads;

means for detecting whether a given thread includes an identifier; and

means for selectively continuing execution of said given thread depending upon whether said identifier is detected.